WHAT IS CLAIMED IS:

1. A parallel process execution method with which a plurality of processors execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, the method comprising the steps of:

(a) setting a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program;

(b) assigning each parallel process of the parallel program to one of the plurality of processors, and starting execution of the assigned parallel processes simultaneously on the plurality of processors; and

(c) stopping the execution of the assigned parallel processes simultaneously on the plurality of processors, when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program.

2. The parallel process execution method according to claim 1, wherein said setting step (a) sets the time allocation ratio to the parallel program by dividing the given cycle period into a plurality of time slots and determining which process to execute in each time slot of the different processors.

3. The parallel process execution method according to claim 2, wherein the processes to be executed in the time slots include interactive processes.

4. The parallel process execution method according to claim 2, further comprising the steps of:

providing a set of criteria beforehand for use in determining what to execute in a free time slot that has no process assigned; and

according to the criteria, selecting at each of the processors which process to be executed in the free time slot.

5. The parallel process execution method according to claim 4, wherein:

the criteria include a throughput-first policy which allows batch processes to run in the free time slot, and a turnaround-first policy which allows no batch process to run in the free time slot; and

said selecting step selects either the throughput-first policy or the turnaround-first policy.

6. The parallel process execution method according to claim 5, wherein the throughput-first policy gives successively lower priorities to interactive processes, non-parallel processes for execution on a

single processor, and parallel processes.

7. 7. The parallel process execution method according to claim 1, wherein the processors are distributed over a plurality of nodes, and the method further comprises the steps of:

causing simultaneous interrupts to the nodes;

sending the received interrupts simultaneously to every processor in the nodes; and

causing the processors to start the cycle period in phase with the interrupts.

8. A multiprocessor computer which employs a plurality of processors to execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, comprising:

a time allocation ratio setting unit that sets a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program; and

a process execution unit that assigns each parallel process of the parallel program to one of the plurality of processors, starts execution of the assigned parallel processes simultaneously on the plurality of processors, and stops the execution of the assigned parallel processes simultaneously on the plurality of processors when the time elapsed since the start of the

parallel processes has reached a point that corresponds to the time allocation ratio.

9. The multiprocessor computer according to claim 8, wherein said time allocation ratio setting unit divides the given cycle period into a plurality of time slots and determines which process to execute in each time slot of the different processors.

10. The multiprocessor computer according to claim 9, wherein:

a set of criteria are provided beforehand for use in determining what to execute in a free time slot that has no process assigned; and

said process execution unit causes each of the processors to select, according to the criteria, which process to execute in the free time slot.

11. The multiprocessor computer according to claim 10, wherein:

the criteria include a throughput-first policy which allows batch processes to run in the free time slot, and a turnaround-first policy which allows no batch process to run in the free time slot; and

each of the processors selects either the throughput-first policy or the turnaround-first policy, according to the criteria,

12. A multiprocessor computer which employs a plurality of processors to execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, the plurality of processors being distributed over a plurality of nodes, the multiprocessor computer comprising:

a coordination controller that generates interrupt notifications to a plurality of nodes simultaneously;

a time allocation ratio setting unit that sets a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program; and

a plurality of process execution units each of which is disposed in the plurality of nodes, assigns each parallel process of the parallel program to one of the plurality of processors, starts execution of the assigned parallel processes simultaneously on the plurality of processors, and stops the execution of the assigned parallel processes simultaneously on the plurality of processors when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio.

13. A parallel process execution program for use with a plurality of processors to execute a plurality of parallel processes produced from a parallel program

together with other processes in a time-shared fashion, the program causing a computer to perform the steps of:

setting a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program;

assigning each parallel process of the parallel program to one of the plurality of processors, and starting execution of the assigned parallel processes simultaneously on the plurality of processors; and

stopping the execution of the assigned parallel processes simultaneously on the plurality of processors, when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program to allocate the given cycle period.

14. A computer-readable medium storing a program for use with a plurality of processors to execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, the program causing the computers to perform the steps of:

setting a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program;

assigning each parallel process of the parallel program to one of the plurality of processors, and

starting execution of the assigned parallel processes simultaneously on the plurality of processors; and

stopping the execution of the assigned parallel processes simultaneously on the plurality of processors, when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program to allocate the given cycle period.